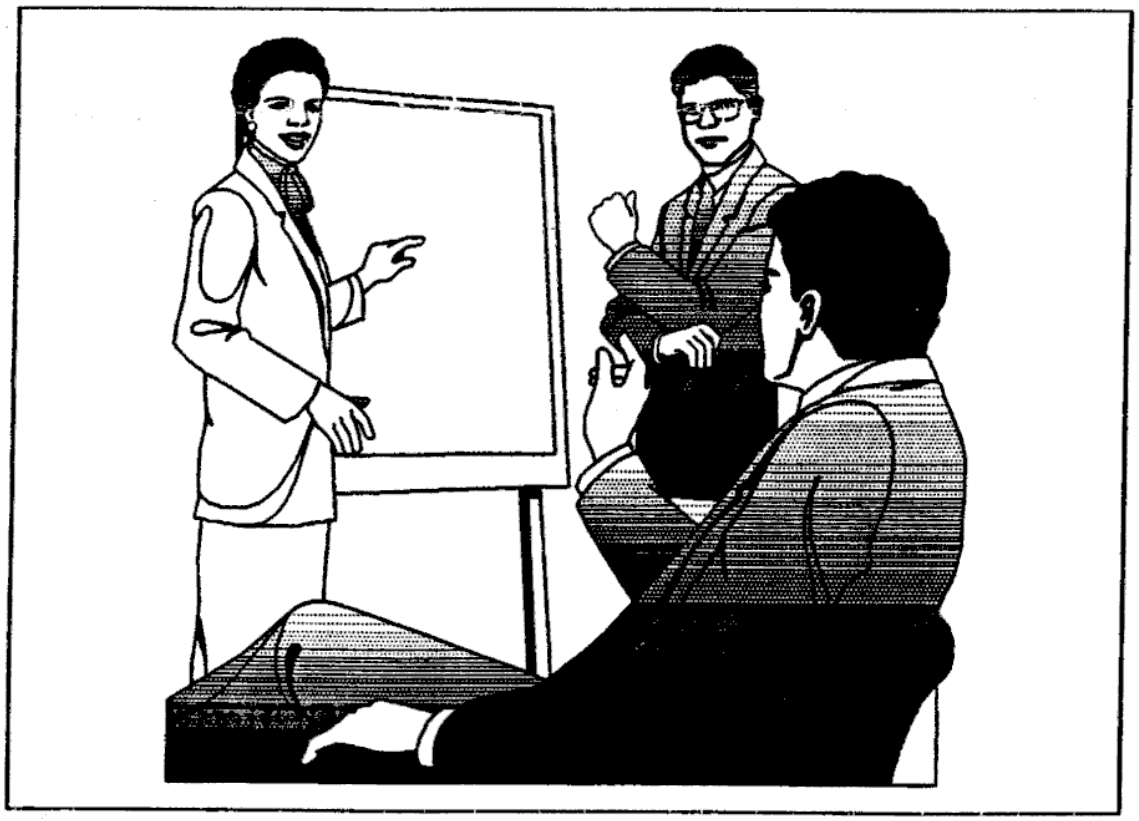
Issue: DECEMBER 1996

LISTing Policy ..... Annual Dues $ 16.00

LISTings

Newsletter of the Long Island Sinclair / Timex Users' Group

NEXT LIST MEETING January 12th, 1997



On e sample copy sent upon receipt of business size SASE. Copies provided on Exchange basis with other Bona fide user groups. We are always looking for articles, programs, reviews, etc to keep members informed and entertained. You maintain full credit and copyright.

Portions of this publication may be reproduced, without need for written consent. Please give credit to LISTing when reprinting any articles. List disclaims responsibility for any damage to your computer as a result of reading any articles in LISTing.

Computing Computing's True Cost

Corporate computer managers have known it for years, but most individual personal computer buyers learn it the hard way: the purchase of one PC is just the tip of the iceberg.

Many families are saving to buy a new computer for the holidays. But the excitement of getting the new plaything can evaporate quickly if one does not account for the myriad additional expenses that it brings. For those who will be receiving a PC from Santa, remember: a computer is a gift that keeps on taking.

Beyond the obvious extra outlay for sales tax, other hidden or camouflaged costs can surprise, disappoint and ultimately dismay the buyer. In general, if a new personal computer costs $2,000.00, it will demand another $2,000.00 worth of software, peripherals and toys.

The budget creep begins on the first day of shopping: First, the advertisement may show a computer with a CD-ROM drive, a big screen monitor, a keyboard and mouse and perhaps a printer, topped with a sign that reads, "Only $1,495.00" or something like that. Grab the magnifying glass. The small print typically advises, "monitor and printer not included". Monitors and printers add at least a couple of hundred dollars each to the system price.

We tend to think poorly of used-car salesmen, but at least they do not ambush us with "Steering wheel and tires extra". Or the macho-looking PC is shown with two headlines, one that reads, "250 Megahertz!" and the other, "Starting at $1,999.00!" Do not assume the two selling points are related. Again, the fine print reveals that the $1,999.00 model uses the runt littermate of the fast chip, or has only half the memory of the bigger model, or has the bargain-bin 4X CD-ROM drive, and so on.

By the time the sales person persuades you that you really want the better components, the extra memory, the modem, the year or two of onsite service, and the surge protector, and the box of diskettes, and the external speakers, and - what the heck - the mousepad, and the wrist rest, and the joystick, the video eyeball gizmo, and the special printer paper, the scanner, the special ergonomic chair and computer desk, the propeller beanie ... Yow! Your budget is in shreds.

Most of these things are subject to my 180-day rule: As soon as you determine you absolutely have to have something, put it on a list and date it. Wait six months. By then, you will have discovered either that you really don't need it or that its price has come down and the bugs have been fixed.

Some extra dollars, however, are well spent at the time the computer is bought. Although many new PCs come with 16 megabytes of system memory, there are a few that still come with 8. Meanwhile, the software companies continue to crank out bloatware, and there are no signs the trend to memory-hogging software is abating.

Fortunately, memory chip prices are very attractive right now. One can add chunks of memory in 8 megabyte increments for about $50.00 each. Consider equipping the new PC with 32 megabytes of RAM while the price is right.

(continued on page 10)

QL CORNER

Last month I stated that I had purchased a clear Black light kit for use as an eprom eraser. The kit was assembled and did NOT do the job of erasing eproms. I gave the kit to an eleven year old boy who lives next door to me. He use will be for locating invisible flurescent minerals. I hope he has better luck with it than I had with it!

I use my QEP eprom programmer quite often, and occasionally I have an error in the program within the eprom. Out with my old eprom eraser which takes approximately twenty-five minutes to erase a 27C512 eprom. This is a long time to erase the entire chip.

I have a pretty good library of articles for all sorts of electronic items. Searching back for some articles which were written approximately 15 years, I stumbled upon a letter from a friend of mine from Canada. He had experimented with various lamps for erasing eproms. Here it is in it's entirety, a simple and inexpensive way to erase eproms. The author does not want to be acknowledged!

Virtually all eproms require a light source of 2537 Angstroms to erase the contents back to its original state of all outputs high or all outputs low. For a long time the only method of getting that type of light source that I new of was to buy a commercial eprom eraser. By the time I found a cheap eprom eraser, I was looking at over one hundred dollars. Since then I've found you can have an eraser for as low as $27.00 (this was January 1981).

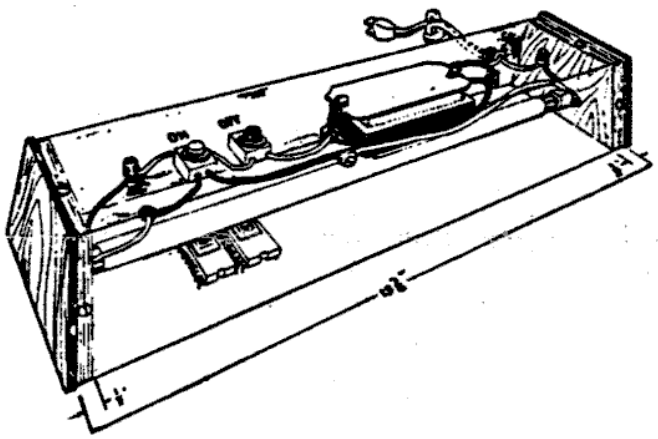
At 2537 Angstroms light does not pass through common glass or the atmosphere very well. This means ordinary mercury vapor ultra-violet sources such as the ones you use for exposing sensitized PC boards or making blueprints do not work when it comes to erasing eproms. Exposing to direct sunlight doesn't seem to work either because of the atmospheric absorption of light rays in this region. I can safely say this because I've tried these methods including "black lights" and plant lights.

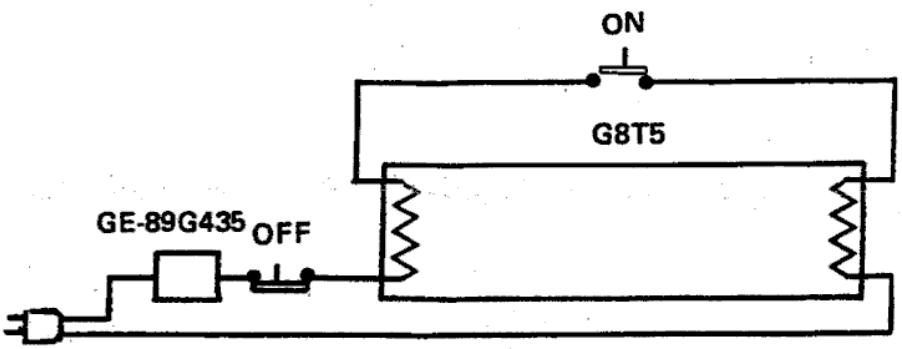
After searching around I finally found the G8T5 bulb (better known as germicidal lamp). The beautiful part of this lamp is that it will fit any standard desk top flourescent lamp that uses a 12" bulb. All you have to do is replace the existing bulb with the G8T5 or equivalent. The bulb isn't cheap so you had better not drop it too many times. The secret to this bulb is quartz tubing instead of glass.

Placed at 3/4" away from the eprom window it will erase the eprom contents in about 18 minutes and even less for a 2716 eproms. Because of varying situations, i.e., age deterioration, dust particles on the eprom window, centering of the light source, etc., you should run a complete "read" of all the addresses on the newly erased eprom to ensure there is not a partial erase on any of the bits. It can be very frustrating to find a partially erased bit at a location after you've keyed in everything correctly up to that point.

Warning! Although light sources in this range of the spectrum may not look very dangerous or even very bright, it can cause skin cancer under continued exposure. At 2537A, it is well beyond the light spectrum you can see but it will still destroy the cells in your eye, eventually causing blindness. The moral of this story is to clearly label a warning on your lamp fixture and don't try to read with this bulb!

For safety reasons or if you don't have a suitable lamp fixture already you can build an inexpensive unit such as shown in the drawing below. It's simply a piece of .032 aluminum folded into an inverted 'U' trough with 3/4" hardwood plywood ends mitered as shown to keep the light from escaping at the ends. The wiring is shown in figure A. Simply center this light source over your eproms, turn it on and come back after the required exposure time. If you're building from scratch, be sure to mount the lamp sockets so that there is a least 3/4" clearance between the bottom of the lamp and the base line. You can even save on sockets by soldering the leads directly to the bulb.





The Eprom

The first memory devices for computers were bi-stable vacuum tube circuits known as 'filp flops' and sonic pulses circulating through a length of metal or a tube of mercury. The first memory device that stayed around awhile was the "core" memory. This used a little donut of ferrite that could be magnetically polarized by a pulse of current in a wire passing through its center. The problem with this was that you could only tell which way it was polarized by giving it another pulse big enough to change its polarization and thus erasing the stored data.

These little donuts (cores) were woven into "mats" with three wires through each core, two for select and one to read the condition. Whenever you read out a bit, you had to rewrite it if you wanted it to be there. In spite of this complexity, core memories became the dominant form till the semiconductor memory replaced it in the early 70's. The first semiconductor memories were like the vacuum type. A bunch of flip flops, but made very small.

In rapid sequence Intel introduced the first SRAM (Static Random Access Memory) and the DRAM accesses the same as the SRAM, but has a simpler memory element. The DRAM accesses the same as the SRAM but has a simpler memory element. It operates by storing the data as a charge on a tiny capacitor. This allowed a smaller memory cell and thus a larger memory matrix for a given chip size, but because of the small size of the storage capacitor, the memory had to be "refreshed" several hundred times a second. That is; whatever state it was in, it had to be rewritten to enhance that state. At first this seemed to many to be a precarious way to store data, but the DRAM proved reliable and is now the most common memory element.

In 1972, very shortly after the introduction of the DRAM, Intel also developed the EPROM (Erasable Programmable Read Only Memory). This device uses a single MOS transistor as the memory element. It is equipped with a "floating gate" that controls whether it is conducting or not. This gate is like a little capacitor plate over the transistor that is completely surrounded by insulating oxide. Because there is normally no conductive path to it, a charge or voltage level will remain indefinitely. The trick is to have a connection for normal use. This is done by making the insulating oxide thin enough, and uniform enough in its thinness, that a higher than normal voltage applied will cause a nondestructive break down or leak that will change the voltage level of the floating gate. If it can be changed both ways it is called an EEPROM (Electrically Erasable PROM) but this proved very difficult and only recently EEPROMs (or E2PROM) become competitive. The single direction gate charge was easier and another way was found to reverse the effect; UV light.

The charge that could be put on the floating gate could be removed by exposing the transistor to 254 nm wavelength UV light. There is nothing magic about 254nm. It just happens to be the easiest wave length to produce with a mercury vapor lamp and conveniently is nearly absent from sunlight and common light sources.

A 27256 EPROM has 262,144 bits of memory like the 256K DRAM, but since it is organized as Byte wide data, there are only 256K/8 = 32K memory locations and only fifteen address lines are needed. With fifteen address lines and 8 data lines at least 23 pins are needed. Add power, and you have 25 pins. There are several other functions needed and these are cleverly combined in the remaining three pins of a 28 pin EPROM package.

One of these functions is to allow the data lines to be used as both in and out. Another is to provide the higher programming voltage needed for writing to the EPROM. Since all memory devices in a system are tied to the same address and data lines, control functions are needed to allow the address information in (usually called "Chip Enable" or CE) and a means to make the data pins connect or not to the computer data lines (usually called "Output Enable or OE).

Originally the small capacity EPROMs had all these control functions on different pins. As more memory was provides on a chip and more address lines were needed, some of these control pins were required to serve multiple purposes.

Thus, in most EPROMS used now, the control functions are a complicated combination of voltage levels on these control pins. The 27256 for example has eight different operating modes all determined by three control pins.

The EPROM started out in a 24 pin package, as originated by Intel. Some other manufacturers made EPROMs in packages of fewer pins, but they have mostly vanished from use and the Intel standard is now almost universally used.

The first EPROM, the 2708, was a 24 pin device and, as mentioned, was a straight forward control method. The 2716 was the next and the added address line meant complicating the control a little. The 2732 was still a 24 pin package and called for still more combined control pin action. With the 2764, the package size increased to 28 pins and simple control again. The 27128 had almost the same control functions

but with the 27256 it got complicated again. Each of these has a little different control method for programming, though the "Read" function is straight forward. Along with this development of larger size memory matrix, methods were found to reduce the programming voltage. It went to 25V on the early units to 21V and now to 12.5V.

As stated earlier, an EPROM is read at full computer speed much as any other memory device. Writing is quite a different situation. The EPROM data can only be changed from "1's" to "0's" by writing. "1" being a high output data line and "0" being a low. High is generally about 4V and low is less than .5V.

Writing an EPROM is slow compared to computer speeds. It requires applying programming voltage and data to the EPROM while the desired address is selected for as much as 50 milliseconds per Byte.

Actually, the way the EPROM's are written is to apply these signals for about one millisecond. Then the mode is switched to "Verify" which allows read out without changing the programming voltage. If the data read out is not the same as that being written, the 1ms write cycle is repeated. This is continued till a proper verify is observed. At that point the EPROM writing program applies the programming data and address, as before, but for a period of time that is some function of the total time it took to get that Byte to absorb the data. For example, Intel suggests the final pulse should be three times as long as the total of all the pulses it took to get a good "Verify" response. This is to change the floating gate well beyond a threshold condition.

The newer EPROMs have the capability of much faster writing. One method is to apply 6V to the Vcc pin (normally 5V) and use 100 microsecond programming pulses. This takes advantage of the fact that most Bytes write much faster than the worst case. Also, operating at Vcc = 6V during programming has something of the same effect as the final "over program" pulse of the other method. When the "Verify" is threshold at Vcc = 6V it is well over threshold at Vcc = 5V.

As mentioned, the EPROM bits can only be written from "1" to "0". Thus, to write an EPROM it is generally necessary to restore the entire memory matrix to "1's" by ultra violet light erasure. If read out in Hex code, the Bytes appear as FF indicating all bits are "1's".

EPROM programmers have the means of applying the address and the data codes to the EPROM and holding them there for the relatively long time necessary. They also have the means of applying the proper program voltage and control codes to suit the particular EPROM being used.

Normally a programmer provides:

Erase check: To check for full erasure.

Verify: To check the stored data.

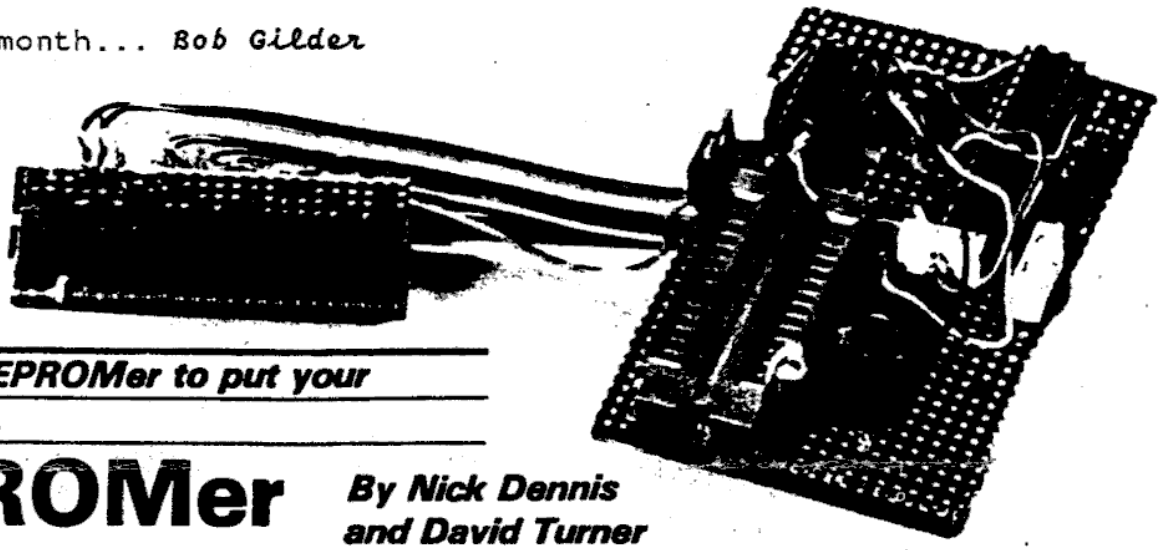
Write: To store a block of data.

The Write" function incorporates the first two also. That is; it will do an Erase check just prior to writing and abort if the EPROM is not blank. However, most programmers allow you to bypass this routine if desired so that just a portion of the EPROM can be written. Ordinarily, on completion of the "Write" cycle, the program does a "Verify" routine just to be sure it does read out properly. The Erase

Check and Verify routines can take less than 10 milliseconds for a 256K EPROM so they are not noticeable relative to the minutes necessary for the "Write" code.

Many thanks for the above information from Intel.

See you next month... *Bob Gilder*



A low-cost EPROMer to put your ZX81 to use.

ZX-EPROMer

By Nick Dennis and David Turner

EPROMS are powerful, inexpensive and familiar components that are found in many circuits popular today. A stand-alone programmer can cost over $1000. A card for your fruit is cheaper, but then you must have a machine to put that card in. Stop using your ZX-81 as a doorstop and cash in on its hidden potential; build this "full-feature" EPROMer and enjoy freedom over EPROMs. The circuit described here will program the popular 2764 type EPROMs; these are under $10 and store 8K, making them a good choice for many applications. Some uses of this device could be:

1. Program any compatible EPROM.(i.e., 2716).
2. Read any ROM, PROM, or EPROM.
3. Static RAM in any unoccupied memory block.
4. A games cartridge system for the ZX-81.
5. An EPROM copier.

How It Works

While we will be looking at programming the 2764, the theory of operation is much the same for other types. It is well worth your while to consult the proper manual before trying a different memory chip. The programming voltages, for example, vary from type to type. When programming the 2764 it needs:

1. Stable data and address lines for the duration of the write.
2. A stable 21 voltsat 50mA at the Vpp pin.
3. The PGM and CS held low for 50mS after 1 and 2 are met.

The OE pin can be tied to CS to allow for verification of each write. Vpp can be either 5 or 21 volts while reading; PGM should be held high when reading. In this circuit we are slipping our 8K EPROM into the memory map in the "never bothered by BASIC" 8-16K block. As such, PEEK and POKE can give us direct access to all locations. This means that all software can be written in slow but friendly BASIC.

IC 1 is a block decoder. It looks at the top three address lines and decodes all memory requests into 8K wide Block Selects (8 outputs x 8K = 64K). The Y0 BS de-echoes the ROM (familiar ZX stuff) while its neighbour Y1 delivers the BS where our EPROM lives.

IC 2 is the trap circuit. If BS is low and RD is high (this can only mean a write to the EPROM) and if the timer output is high, then a WAIT is issued to the Z80. This has the unadvertised effect of freezing the Z80's lines until released. The WAIT is also inverted and after a slight delay (C3) passed to the first timer's + ve trigger. Each timer has two outputs, Qbar and + Q. Because the 2764 is programmed with a high-low-high sequence PGM is connected to Qbar. The Q output is applied to the second timer's -ve trigger. After 50 milliseconds these outputs both flip, thus ending the programming and triggering the second timer. The second timer prevents the LS10 from retrapping this same write. But what about REFRESHing the dynamic RAMs? Even though D-RAM is usually REFRESHed every few milliseconds, experience has shown that good data integrity is maintained with occasional periods much longer than 50mS.

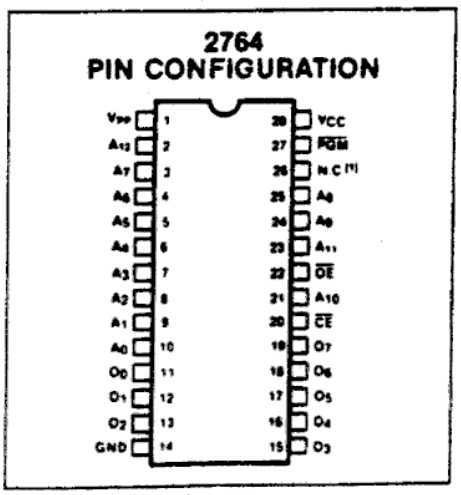


Fig. 1 The pinouts of the 2764 EPROM.

Hooking'er Up

This project is fairly straightforward and does not require a printed circuit board. Veroboard is a good choice for most projects like this. Connecting the veroboard to a ZX-connect with coloured ribbon cable will keep wiring simple. As a bonus, not every edge-connect line has to be brought out to the board.

A ZX-connect can be made by cutting a 50 pin 1/10 inch standard edge-connect to length. The key-way tab is made from veroboard. The expansion strip can also be made from two pieces of veroboard sanded to half thickness and glued back to back. An easier approach is to cut the strip from an existing circuit board. Our strip came from Zebra systems but we have seen many ideal bits of fibreglass in our favorite surplus store. The 1/10 inch spacing is very common.

All this is necessary because the EPROMer must be the first device plugged into the expansion port. The Memory Packs alter the high order address lines (to be stackable) and this definitely won't do. Many peripherals are memory mapped in the 8-16 K range and as such are not compatible with this gadget. To be able to use a 64K pack with the EPROMer you must be able to de-select the 8-16 K range. Most have this feature. Any I/O mapped

device (e.g., the ZX printer) is cool.

If the expansion strip is not possible for some reason then bring any altered address line (typically A14) directly from inside the computer to IC1. This is low-tech and not too neat, but will still do the trick.

Personality The use of several IC sockets as "personality" strips is recommended but not vital. The idea here is to be able to make modifications to control hookups without soldering. A look at the pinouts for most popular ICs reveals that many lines always connect to the same pins while others change from type to type. Having the top few address lines and most of the control lines jumpered through a row of IC sockets will allow a lot of scope for testing, adjustment and hacking.

The Data lines and A0 to A10 are wired directly to the socket. Solder the control lines as the circuit comes together. WAIT, RD, MREQ go directly from the circuit to the ZX-connect. A good "personality" candidate is WR . While not used in the circuit, it allows the use of static RAMs in the socket.

It is worth noting that no points are awarded for a cramped, tiny board. While keeping all wire lengths as short as possible is always a good idea, never paint yourself into a corner when building around a flexible circuit like this.

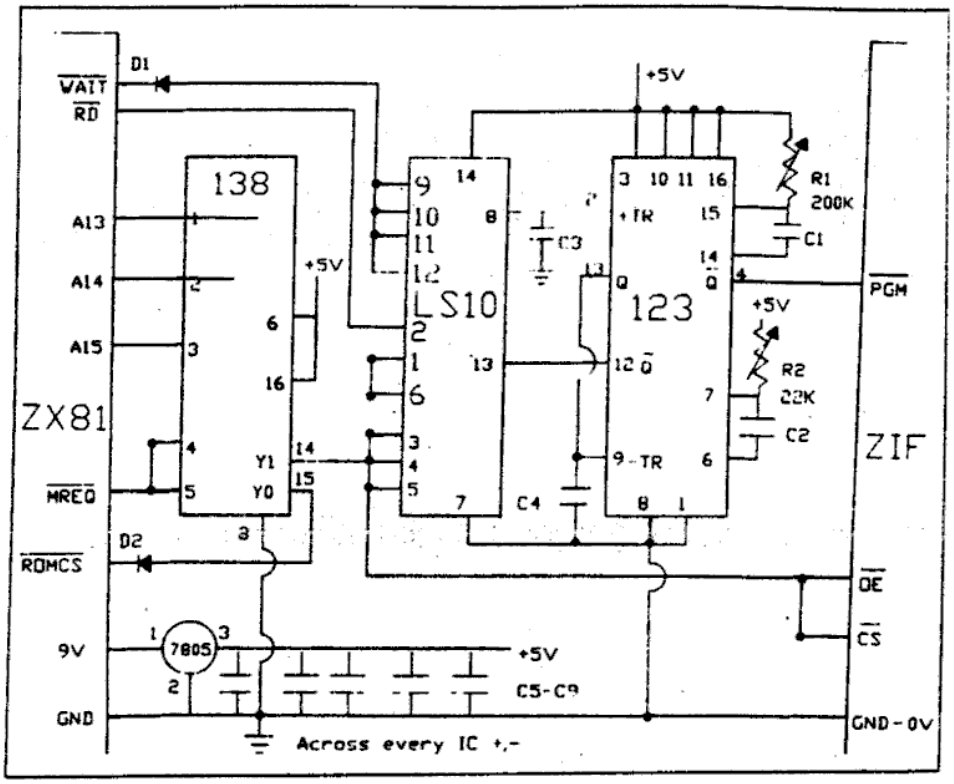


Fig. 2 The complete circuit of the ZX EPROMer

Testing and Adjustment

When your board is all assembled and well-checked over for solder bridges, it is time to power it up. Connect it directly to the ZX edge-connect and (with no EPROM) apply the power. If the cursor does not appear re-recheck all wiring.

If it does, try a POKE to anywhere in the block, e.g., POKE 10000,0. A screen flicker at this time tells you that all is well. This is the Z80 "freezing".

If you are unable to get a screen flicker, test the trap by momentarily grounding 3, 4, and 5 of the LS 10 (disconnect them from LS 138 first). A screen flicker now means that the problem is in the block decoding. If you are still having problems, read the section "Gotcha".

But you were careful and you got the screen flicker right away. Centre R1 and R2 and enter the program, except for line 120. The idea here is simple: while 50 milliseconds is hard to time with a stopwatch, 50 seconds is pretty easy. Disconnect the LS138 from the LS10 and (no flicker) RUN the program. This is approximately 70 seconds in SLOW mode. Reconnect the LS138 to the LS10 and re-RUN the program. For 1000 loops the "flicker" should add 50 seconds to the "no flicker" RUN time, giving a total time of 120 seconds. Adjust R1 until this is

Pres.	Harvey Rait
Vice Pres.	Bob Gilder
Tresurer	Robert Malloy
Cor. Secy.	John Pazmino
Assoc. Editors	Fred Stern
	Harvey Rait
Publisher	Bob Gilder
Libr.	Tom Skapinski

Please send all inquiries and submissions (including dues) to:

L.I.S.T.
Mr. Harvey Rait
5 Peri Lane,
Valley Stream, N. Y. 11581

COMING EVENTS: The next L.I.S.T. meeting will be Sunday, Ø1/12/97 at 2 P.M. at the home of Harvey Rait (see address above).

exactly right. Now put a drop of your favourite nail polish on RI's arm. By now the use of the "personality" strip should be apparent. Without it this setup stage can be a hassle.

R2 is not too critical. If the system "hangs" on the first write, then increase its value. Add line 120, delete line 30, CLEAR the variable area and SAVE the program to tape. DEST and the number of LOOPs will change for different applications, but the theory of operation and the timing set by RI will remain the same.

Programming

To simplify programming we will be using BASIC. Because the socket is "memory-mapped", PEEK and POKE will give us access to the EPROM. PEEK behaves normally, while POKE is extended by the circuit to meet the 50 millisecond requirement. Remember, POKEs are permanent! To erase any one location you must erase the entire EPROM.

The EPROM can be programmed in many ways or in any order. Most assemblers have a Block Transfer or MOVE function, but usually without verification. The easy solution is to move the object code into a string variable and then type in the minimum BASIC program. With 16K of memory there is still ample room for most assemblers (4K-7K), a few lines of BASIC, and 8K of data in a variable.

To copy an existing EPROM, for example, transfer its contents to an array, save everything to tape, power down, install the blank, reload and then program.

The BASIC given here is only to show how simple it is to program an EPROM. Feel free to include Menus, features, bells and whistles if you like. Your program should:

1. Verify that all locations to be programmed are 255 ($FF) before writing. You can make a one into a zero but not vice versa.
2. Verify all start/finish values with the user.
3. Verify that the correct value has been written. If an error occurs it is not likely that the machine can correct itself.

Vpp drifting or poor socket contacts are the initial suspects so STOP if even one write will not verify.

Mods

A look at the pinouts of these common chips reveals how similar they all are. For example, even if we can't program a 2732 type, it is still possible to READ its contents. The contents of the 2732 can then be put in half of a 2764. When installing the "double 32" connect the highest address line to either 0 or 5 volts depending on which half you want to use. These devices are cheap enough to make this method the easiest way to get around some tricky circuit construction.

Gotcha

The most common problem with project or kit failure is faulty soldering. Before powering up, go over the entire board with a magnifying glass. Tiny wisps of solder, while barely visible to the eye, can cause expensive damage. Going over the board thoroughly is time well spent. If your project will not work, examine each section independently. Remove the other ICs from the board, and using the "How it Works" section, trace the circuit's operation. With only the LS138 installed, PEEKing 10000 should return 255. Any other result means that the ROM echo is not suppressed; therefore, re-check the 138. The trap and the timer can be checked statically, making them much simpler to debug.

The next most likely trouble source is the ZIF socket. A good socket will cost half of the total cost of the project, and is worth every cent. The cheapies do not guarantee the good connection that must be maintained.

When programming, bear several things in mind. The 21 volts needed to program an EPROM is *fatal* to all other TTL circuits. If this voltage varies by more than a half a volt either way, the write may not take. It is important that the chosen supply can deliver at least 50mA with good regulation. An on-card 5 volt regulator and a decoupling capacitor across every IC also helps maintain the accuracy needed for programming. These capacitors must not be eliminated or you will be plagued with unpredictable results.

```
    5 REM THE ZX EPROMER
          BY D.TURNER + N.DENNIS

   10 REM THIS ASSUMES THAT THE
          CODE TO BE PROGRAMMED
          IS ALREADY STORED IN A
          VARIABLE CALLED D$.

   15 REM IT SHOULD TAKE 50 SEC
          LONGER TO EXECUTE 1000
          LOOPS WITH THE EPROMER
          ACTIVE (SCREEN FLICKER)
          THEN WITHOUT.
   20 REM TYPICAL TIME FOR 1000
          LOOPS OF THE PROGRAM
          IS 70 SECONDS.
          THIS TIME DOES NOT
          INCLUDE LINE 120 WHICH
          WOULD STOP THE TEST.

   25 REM LINE 30 IS A DUMMY FOR
          TESTING ONLY. DELETE IT
          AFTER TIMING OR IT WILL
          DESTROY THE DATA IN D$.

   30 DIM D$(1000)
   50 REM
   90 LET DEST=8192
   99 SLOW
  100 FOR I=1 TO 1000
  110 POKE DEST+I,CODE D$(I)
  120 IF PEEK (DEST+I) <>CODE D$(I
) THEN GOTO 999
  150 NEXT I
  200 STOP
  999 PRINT "VERIFICATION ERROR A
T ",I
```

Fig. 3 A simple BASIC listing for programming and compatible EPROM.

PARTS LIST

I.C.s

I.C.1 ..... 74LS 138 block decoder
I.C.2 ..... 74LS 10 triple Nand
I.C.3 ..... 74LS 123 dual timer
I.C.4 ..... 7805 voltage regulator

Resistors, Capacitors

R1 ..... 200K 10 turn mini-trim type pot.
R2 ..... 22K trim pot.
C1 ..... 1.0 uF tantalum cap.
C2 ..... 500 pF cap.
C3, C4 ..... .005 uF cap
C5-C9 ..... 1 uF (decoupling)

Misc.

D1, D2 ..... 1N4009 or approx. equivalent diodes
ZX edge connect and expansion strip
I.C. sockets (4-5 with personality)
Veroboard, Ribbon cable
A GOOD ZIP (Zero Insertion Force) Socket
21 Volt power supply ..... 29 Volt and 21.5 Volt (AA) batteries will work just fine if fresh.

Software is likely to be the biggest additional expense over the life of the personal computer. This is the reason Bill Gates is the richest man in America. Many computers come from the factory loaded with software, in some cases dozens of programs with an advertised value of $1000.00 or more. In truth, many of the programs are likely to be demonstration, outdated or "light" versions that lack key features, so take the monetary claims lightly. Then add to the budget to cover the cost of upgrades.

It is not unheard of for a family to spend the equivalent of the computer's purchase price for software over the life of the machine. High-end productivity packages can cost hundreds of dollars, and the Cool Game of the Month always seems to cost $50.00 or more. All that software has to be backed up. The downside of having a gigabyte hard drive is the cheapest way to back it up is on megabyte diskettes. To save time and sanity, not to mention huge stacks of diskettes, consider getting a Zip drive (about $150.00) or one of its removeable cartridge drive cousins.

And there is the Internet, which turns your cash into electrons. If your computer does not come with a 28.8-kilobyte-per-second modem, or something faster, plan to spend about $150.00 or more to add one. These days, a computer without a modem is seriously underequipped. Once the modem is installed, plan to budget at least $20.00 a month for Internet access.

And then there are the hidden costs of the telephone lines. If some family members plan to take advantage of unlimited Internet access, other family members may be frustrated when the phone line is constantly busy. A second phone line will ease the disputes, but again, the phone company's fingers will go walking through your wallet. The unkindest cost of all is the long-distance charges spent for waiting on line to talk with technical support.

The list goes on. There will be spare cables to buy at $10.00 to $15.00 each, at least one exotic adapter for a couple of dollars, a few subscriptions, membership dues for a local computer club and the bribes paid to the neighbor's high-schooll kid to debug your mysterious Windows.INI files. In the final accounting, when you access how mush you have spent, the extra money for the "Dummies" books will seem quite appropriate.

By Peter H. Lewis

ATTENTION LIST Subscribers: When it is time to renew your membership, (look at your mailing label), please make out your check to Harvey Rait, LIST President or to Robert Malloy, Treasurer. PLEASE DO NOT MAKE OUT YOUR CHECK to LIST. Our bank requires a large amount of money in a savings account in order to cash checks. THANK YOU!

Harvey Rait
5 Peri Lane,
Valley Stream, NY 11581

Robert Malloy
412 Pacific Street,
Massapequa Park, NY 11762

Due to rising postage costs outside of the United States, we must raise our annual dues accordingly:

CANADA and MEXICO $17.50 US, and the rest of the world $24.00 US.

Bob Malloy, LIST Treasurer
